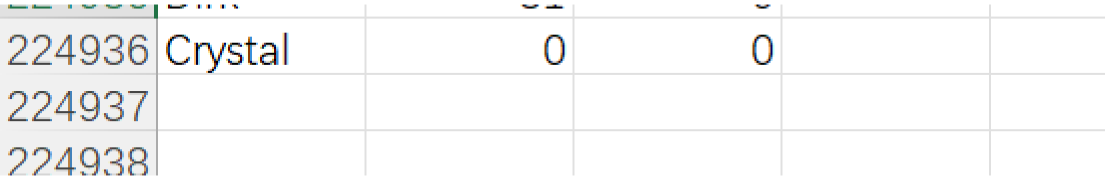
1.

```
MATCH (user:UserNode)-[:Review]->(:ReviewNode)-[:Reviewed]->(business:BusinessNode)
WITH user, COUNT(distinct business) as count
WHERE count>5
RETURN user.name, user.funny, user.fans, count
```



224936	Crystal	0	0	
224937				
224938				

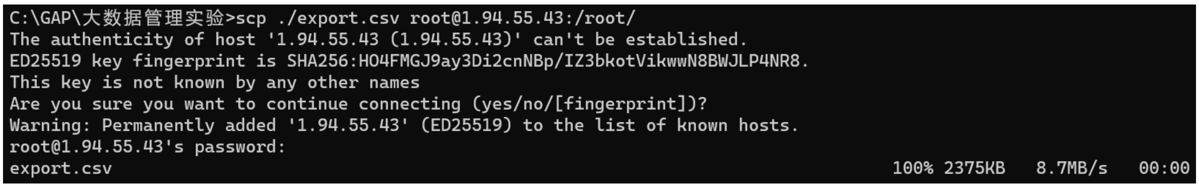
2.

将1得到的结果导入MongoDB，并使用该表格数据，统计其中所有出现的用户名及该用户名对应的出现次数，并按照出现次数降序排序,使用aggregate实现

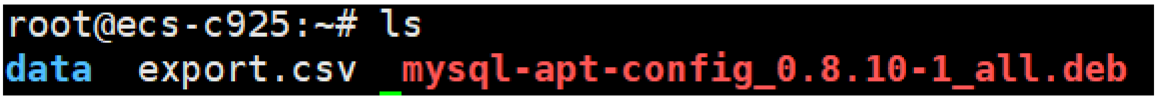
1）从Neo4j的查询中导出csv文件（export.csv）

2）在mongodb新建集合from_neo4j，将csv文件导入集合

C:\GAP\大数据管理实验>scp ./export.csv root@1.94.55.43:/root/



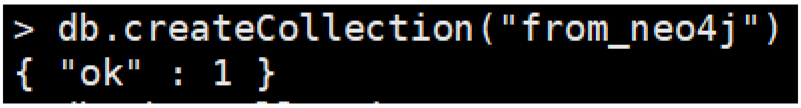
```
C:\GAP\大数据管理实验>scp ./export.csv root@1.94.55.43:/root/
The authenticity of host '1.94.55.43 (1.94.55.43)' can't be established.
ED25519 key fingerprint is SHA256:HO4FMGJ9ay3Di2cnNBp/IZ3bkotVikwwN8BWJLP4NR8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '1.94.55.43' (ED25519) to the list of known hosts.
root@1.94.55.43's password:
export.csv                                                                    100% 2375KB   8.7MB/s   00:00
```



```
root@ecs-c925:~# ls
data  export.csv  mysql-apt-config_0.8.10-1_all.deb
```

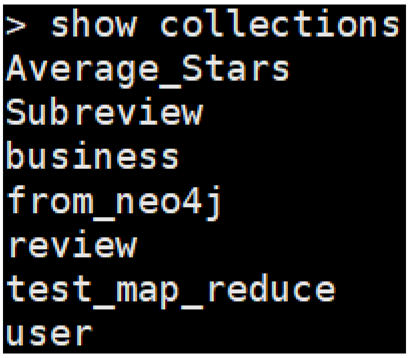
然后启动mongo，选择yelp数据库，创建一个新的集合from_neo4j

db.createCollection("from_neo4j")



```
> db.createCollection("from_neo4j")
{ "ok" : 1 }
```

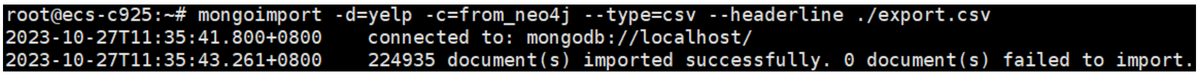
使用show collections查看当前集合：



```
> show collections
Average_Stars
Subreview
business
from_neo4j
review
test_map_reduce
user
```

退出mongoDB，回到主目录，把数据导入到mongoDB中yelp数据集的from_neo4j集合中。

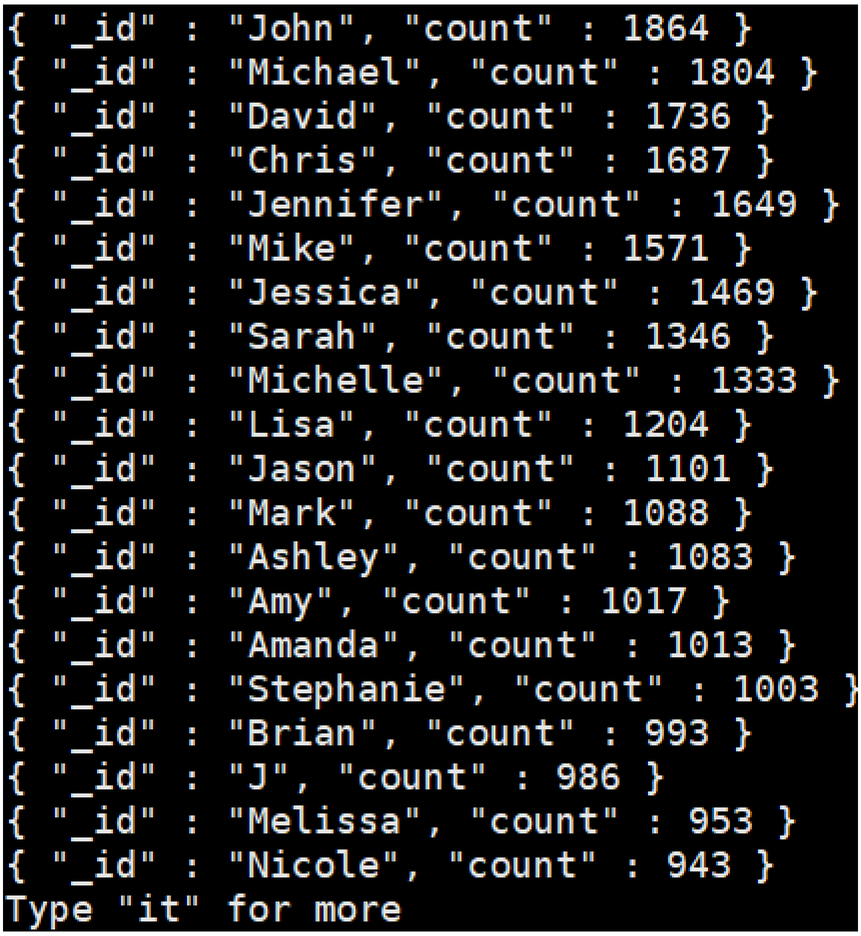
mongoimport -d=yelp -c=from_neo4j --type=csv --headerline ./export.csv



```
root@ecs-c925:~# mongoimport -d=yelp -c=from_neo4j --type=csv --headerline ./export.csv
2023-10-27T11:35:41.800+0800    connected to: mongodb://localhost/
2023-10-27T11:35:43.261+0800    224935 document(s) imported successfully. 0 document(s) failed to import.
```

3）统计其中所有出现的用户名及该用户名对应的出现次数，并按照出现次数降序排序。

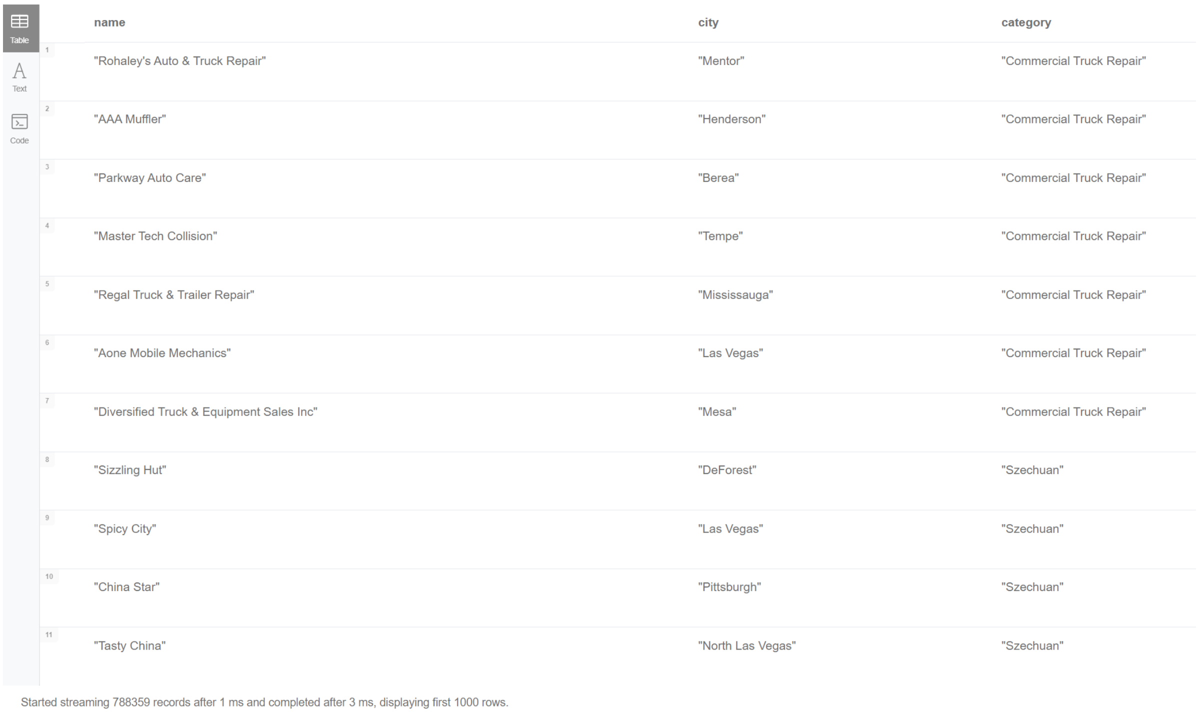
db.from_neo4j.aggregate([
{$group:{_id:'$u.name', count:{$sum:1}}},
{$sort:{count: -1}}
])



```
{ "_id" : "John", "count" : 1864 }
{ "_id" : "Michael", "count" : 1804 }
{ "_id" : "David", "count" : 1736 }
{ "_id" : "Chris", "count" : 1687 }
{ "_id" : "Jennifer", "count" : 1649 }
{ "_id" : "Mike", "count" : 1571 }
{ "_id" : "Jessica", "count" : 1469 }
{ "_id" : "Sarah", "count" : 1346 }
{ "_id" : "Michelle", "count" : 1333 }
{ "_id" : "Lisa", "count" : 1204 }
{ "_id" : "Jason", "count" : 1101 }
{ "_id" : "Mark", "count" : 1088 }
{ "_id" : "Ashley", "count" : 1083 }
{ "_id" : "Amy", "count" : 1017 }
{ "_id" : "Amanda", "count" : 1013 }
{ "_id" : "Stephanie", "count" : 1003 }
{ "_id" : "Brian", "count" : 993 }
{ "_id" : "J", "count" : 986 }
{ "_id" : "Melissa", "count" : 953 }
{ "_id" : "Nicole", "count" : 943 }
Type "it" for more
```

3.

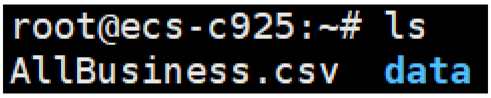
MATCH (business:BusinessNode)-[:IN_CATEGORY]->(c:CategoryNode)
RETURN business.name as name, business.city as city, c.category as category



	name	city	category
1	"Rohaley's Auto & Truck Repair"	"Mentor"	"Commercial Truck Repair"
2	"AAA Muffler"	"Henderson"	"Commercial Truck Repair"
3	"Parkway Auto Care"	"Berea"	"Commercial Truck Repair"
4	"Master Tech Collision"	"Tempe"	"Commercial Truck Repair"
5	"Regal Truck & Trailer Repair"	"Mississauga"	"Commercial Truck Repair"
6	"Aone Mobile Mechanics"	"Las Vegas"	"Commercial Truck Repair"
7	"Diversified Truck & Equipment Sales Inc"	"Mesa"	"Commercial Truck Repair"
8	"Sizzling Hut"	"DeForest"	"Szechuan"
9	"Spicy City"	"Las Vegas"	"Szechuan"
10	"China Star"	"Pittsburgh"	"Szechuan"
11	"Tasty China"	"North Las Vegas"	"Szechuan"

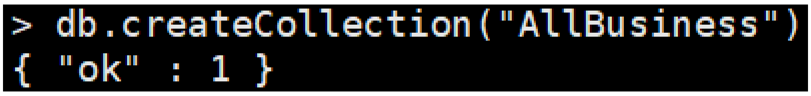
Started streaming 788359 records after 1 ms and completed after 3 ms, displaying first 1000 rows.

然后类似于上面第2题的步骤：



```
root@ecs-c925:~# ls
AllBusiness.csv  data
```

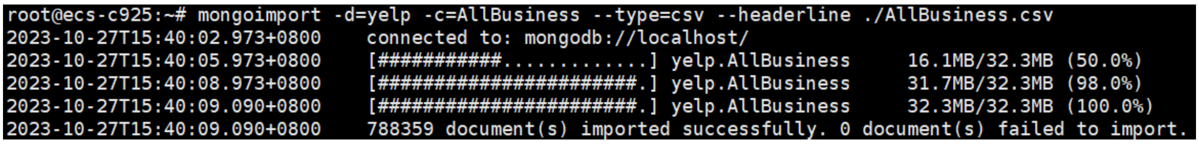
db.createCollection("AllBusiness")



```
> db.createCollection("AllBusiness")
{ "ok" : 1 }
```

退出mongoDB，回到主目录，把数据导入到mongoDB中yelp数据集的AllBusiness集合中。

mongoimport -d=yelp -c=AllBusiness --type=csv --headerline ./AllBusiness.csv



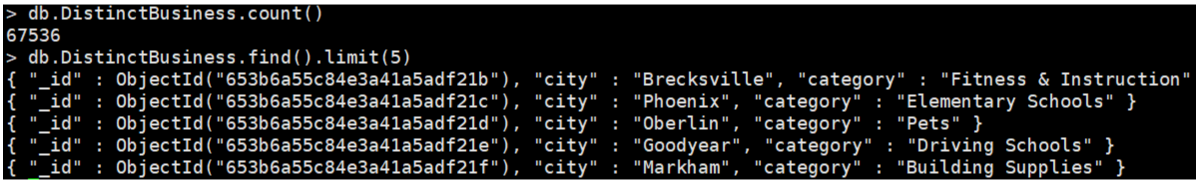
```
root@ecs-c925:~# mongoimport -d=yelp -c=AllBusiness --type=csv --headerline ./AllBusiness.csv
2023-10-27T15:40:02.973+0800    connected to: mongodb://localhost/
2023-10-27T15:40:05.973+0800    [###########............] yelp.AllBusiness     16.1MB/32.3MB (50.0%)
2023-10-27T15:40:08.973+0800    [######################.] yelp.AllBusiness     31.7MB/32.3MB (98.0%)
2023-10-27T15:40:09.090+0800    [#######################.] yelp.AllBusiness     32.3MB/32.3MB (100.0%)
2023-10-27T15:40:09.090+0800    788359 document(s) imported successfully. 0 document(s) failed to import.
```

接下来使用aggregate对AllBusiness去重，仅保留城市、商铺类型。首先创建一个集合用于保存结果

db.createCollection("DistinctBusiness")

db.AllBusiness.aggregate([{ $group: { *id: { city: '$city', category: '$category' } } }]).forEach((item) => { db.DistinctBusiness.insert(item.*id) })

查看结果：



```
> db.DistinctBusiness.count()
67536
> db.DistinctBusiness.find().limit(5)
{ "_id" : ObjectId("653b6a55c84e3a41a5adf21b"), "city" : "Brecksville", "category" : "Fitness & Instruction"
{ "_id" : ObjectId("653b6a55c84e3a41a5adf21c"), "city" : "Phoenix", "category" : "Elementary Schools" }
{ "_id" : ObjectId("653b6a55c84e3a41a5adf21d"), "city" : "Oberlin", "category" : "Pets" }
{ "_id" : ObjectId("653b6a55c84e3a41a5adf21e"), "city" : "Goodyear", "category" : "Driving Schools" }
{ "_id" : ObjectId("653b6a55c84e3a41a5adf21f"), "city" : "Markham", "category" : "Building Supplies" }
```

将结果导出到服务器主目录下的result.csv中。

mongoexport -d yelp -c DistinctBusiness --type=csv --fields city,category --out result.csv

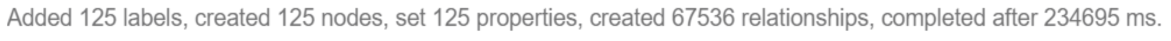
然后将其放在neo4j安装目录的import下

cd ~/neo4j-community-4.0.9/import

cp /root/result.csv ./

将去重后的结果导入Neo4j中的新库result中，完成（City-[Has]->Category）图谱的构建。

LOAD CSV WITH HEADERS FROM "file:///result.csv" AS f
MERGE (c:CityNode {city: COALESCE(f.city, "")})
MERGE (a:CategoryNode {category: COALESCE(f.category, "")})
CREATE (c) -[:Has]-> (a)



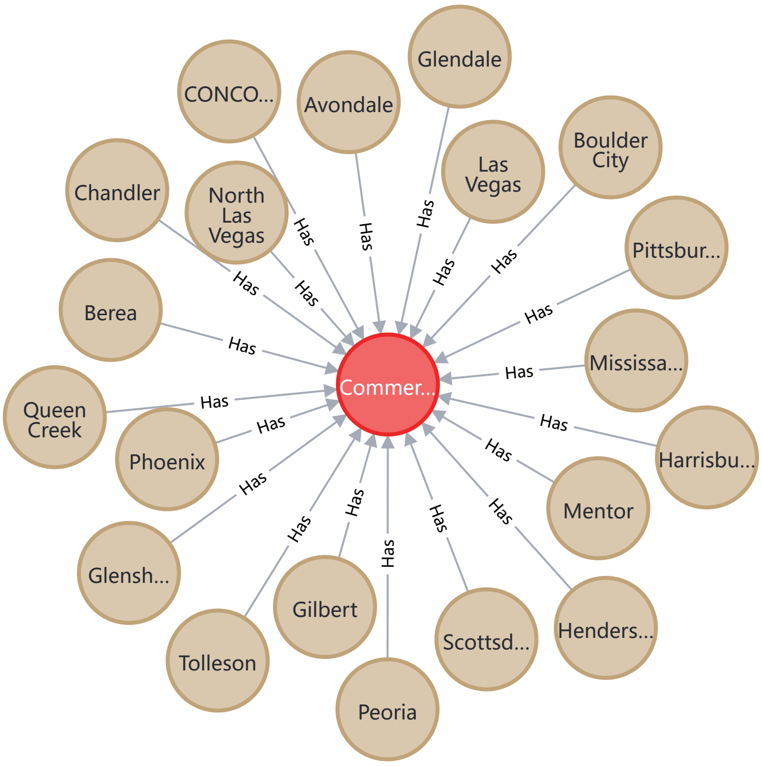
Added 125 labels, created 125 nodes, set 125 properties, created 67536 relationships, completed after 234695 ms.

最后查看City-[Has]->Category图谱

MATCH p=()-[r:Has]->()

RETURN p

LIMIT 20


Commer...
Glendale
CONCO...
Avondale
Boulder City
Las Vegas
Chandler
North Las Vegas
Pittsbur...
Berea
Mississa...
Queen Creek
Phoenix
Harrisbu...
Mentor
Glensh...
Gilbert
Tolleson
Scottsd...
Henders...
Peoria
Has